

iCOMPEL V8.1.0

Technical Reference



Customer Support Information

Order toll-free in the U.S.: Call 877-877-BBOX (outside U.S. call 724-746-5500)
FREE technical support 24 hours a day, 7 days a week: Call 724-746-5500 or fax 724-746-0746
Mailing address: Black Box Corporation, 1000 Park Drive, Lawrence, PA 15055-1018
Web site: www.blackbox.com • E-mail: info@blackbox.com

Federal Communications Commission and Industry Canada Radio Frequency Interference Statements

This equipment generates, uses, and can radiate radio-frequency energy, and if not installed and used properly, that is, in strict accordance with the manufacturer's instructions, may cause interference to radio communication. It has been tested and found to comply with the limits for a Class A computing device in accordance with the specifications in Subpart B of Part 15 of FCC rules, which are designed to provide reasonable protection against such interference when the equipment is operated in a commercial environment. Operation of this equipment in a residential area is likely to cause interference, in which case the user at his own expense will be required to take whatever measures may be necessary to correct the interference.

Changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate the equipment.

This digital apparatus does not exceed the Class A limits for radio noise emission from digital apparatus set out in the Radio Interference Regulation of Industry Canada.

Le présent appareil numérique n'émet pas de bruits radioélectriques dépassant les limites applicables aux appareils numériques de la classe A prescrites dans le Règlement sur le brouillage radioélectrique publié par Industrie Canada.

Instrucciones de Seguridad (Normas Oficiales Mexicanas Electrical Safety Statement)

1. Todas las instrucciones de seguridad y operación deberán ser leídas antes de que el aparato eléctrico sea operado.
2. Las instrucciones de seguridad y operación deberán ser guardadas para referencia futura.
3. Todas las advertencias en el aparato eléctrico y en sus instrucciones de operación deben ser respetadas.
4. Todas las instrucciones de operación y uso deben ser seguidas.
5. El aparato eléctrico no deberá ser usado cerca del agua- por ejemplo, cerca de la tina de baño, lavabo, sótano mojado o cerca de una alberca, etc..
6. El aparato eléctrico debe ser usado únicamente con carritos o pedestales que sean recomendados por el fabricante.
7. El aparato eléctrico debe ser montado a la pared o al techo sólo como sea recomendado por el fabricante.
8. Servicio El usuario no debe intentar dar servicio al equipo eléctrico más allá a lo descrito en las instrucciones de operación. Todo otro servicio deberá ser referido a personal de servicio calificado.
9. El aparato eléctrico debe ser situado de tal manera que su posición no interfiera su uso. La colocación del aparato eléctrico sobre una cama, sofá, alfombra o superficie similar puede bloquea la ventilación, no se debe colocar en libreros o gabinetes que impidan el flujo de aire por los orificios de ventilación.
10. El equipo eléctrico deber ser situado fuera del alcance de fuentes de calor como radiadores, registros de calor, estufas u otros aparatos (incluyendo amplificadores) que producen calor.
11. El aparato eléctrico deberá ser conectado a una fuente de poder sólo del tipo descrito en el instructivo de operación, o como se indique en el aparato.
12. Precaución debe ser tomada de tal manera que la tierra fisica y la polarización del equipo no sea eliminada.
13. Los cables de la fuente de poder deben ser guiados de tal manera que no sean pisados ni pellizcados por objetos colocados sobre o contra ellos, poniendo particular atención a los contactos y receptáculos donde salen del aparato.
14. El equipo eléctrico debe ser limpiado únicamente de acuerdo a las recomendaciones del fabricante.
15. En caso de existir, una antena externa deberá ser localizada lejos de las lineas de energia.
16. El cable de corriente deberá ser desconectado del cuando el equipo no sea usado por un largo periodo de tiempo.
17. Cuidado debe ser tomado de tal manera que objetos liquidos no sean derramados sobre la cubierta u orificios de ventilación.
18. Servicio por personal calificado deberá ser provisto cuando:
 - A: El cable de poder o el contacto ha sido dañado; u
 - B: Objetos han caído o líquido ha sido derramado dentro del aparato; o
 - C: El aparato ha sido expuesto a la lluvia; o
 - D: El aparato parece no operar normalmente o muestra un cambio en su desempeño; o
 - E: El aparato ha sido tirado o su cubierta ha sido dañada.

Trademarks Used in this Manual

Black Box and the Double Diamond logo are registered trademarks of BB Technologies, Inc.

Any other trademarks mentioned in this manual are acknowledged to be the property of the trademark owners.

All rights reserved. No parts of this work may be reproduced in any form or by any means - graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems - without the written permission of the publisher.

Products that are referred to in this document may be either trademarks and/or registered trademarks of the respective owners. The publisher and the author make no claim to these trademarks.

While every precaution has been taken in the preparation of this document, the publisher and the author assume no responsibility for errors or omissions, or for damages resulting from the use of information contained in this document or from the use of programs and source code that may accompany it. In no event shall the publisher and the author be liable for any loss of profit or any other commercial damage caused or alleged to have been caused directly or indirectly by this document.

Printed: February 2011

We're here to help! If you have any questions about your application or our products, contact Black Box Tech Support at **724-746-5500** or go to **blackbox.com** and click on "Talk to Black Box." You'll be live with one of our technical experts in less than 30 seconds.

Table of Contents

1.	Introduction	7
2.	Prerequisites for XML API.....	9
3.	XML API version 2.....	11
3.1	Capabilities	11
3.2	CAP_PLAYER_LOCAL_INFO_1	12
3.2.1	Get All Player Local Information	12
3.2.2	Get Player Local Information	13
3.2.3	Set Player Local Information	14
3.3	CAP_ADVANCE_TO_1	15
3.4	CAP_AD_HOC_ITEM_1	16
3.4.1	Ad hoc Table	17
	- Get Ad hoc Table Info.....	17
	- Set Ad hoc Table Info.....	19
3.4.2	Ad hoc Text	20
	- Get Ad hoc Text Info.....	20
	- Set Ad hoc Text Info.....	21
3.4.3	Ad hoc TV	22
	- Get Ad hoc TV Info.....	22
	- Set Ad hoc TV Info.....	24
3.4.4	Ad hoc Radio	24
	- Get Ad hoc Radio.....	25
	- Set Ad hoc Radio.....	26
3.5	Shared Return Codes	27
3.6	Response List	27
4.	XML API version 1.....	30
4.1	External XML API	30
4.1.1	Channel Manager	30
	- Channel Manager Status.....	30
	- Get Channel Manager Report.....	31
	- Channel Publish.....	33
	- Channel Activate.....	33
4.1.2	Schedule	34
	- Schedule Status.....	34
	- Activate Schedule.....	34
4.2	HTML Player Services	35
4.2.1	Get RS-232 Buffer	35
4.2.2	Transmit RS-232 Data	36
4.2.3	HTTP GET Request	37
4.2.4	HTTP POST Request	38
4.3	Sample Code	38
	Index	41

Part



Introduction

1. Introduction

This technical reference is intended for Web developers and content designers with basic knowledge of web technologies. The purpose is to help you get most out of iCOMPEL Digital Signage platform improving efficiency and accuracy of digital media played and more flexible control over what, when and how media is played on the Black Box digital signage network.

From iCOMPEL version 8.1.0, the XML API has been revamped to provide a consistent and easy to use interface. The API's are grouped into capabilities with each capability providing a specific operation in a specific way. Each capability also locks the schema for that particular operation. Capabilities can therefore be used to verify if a given content can work well with a given version of iCOMPEL.

Part



Prerequisites for XML API

2. Prerequisites for XML API

XML commands should be posted to **/XML2** to make use of the new interface using credentials of a user with **'XML Control'** access.

The new interface is based on capabilities that locks down a schema for each operation supported in that capability.

If a iCOMPEL claims to have a certain capability, it can be safely assumed that the XML commands will be accepted by that version.

There are two types of capabilities:

- Read-only, that just query data from iCOMPEL, and
- State-changing, that change any data in the iCOMPEL

The new interface supports serial execution of multiple commands with a single invocation using a command list with the constraint that read-only and state-changing capabilities cannot be mixed in a single list.

The `'Content-type'` header must be set to `'text/xml'`. The XML version 1 API's will continue to work on **/XML** but will be deprecated in future.

Part



XML API version 2

3. XML API version 2

3.1 Capabilities

Capabilities in iCOMPEL XML interface provide a mechanism to determine if a given version of the product supports a specific set of API's.

Applications can request the currently available capabilities by using an assured backward compatible interface.

Syntax

The syntax of this message is:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<command_list>
  <get_all_capabilities command_id="get cmd caps id"/>
</command_list>
```

XML Node	Notes
<command_list>	Root element for all operations that can be requested
<get_all_capabilities>	Command name
command_id	A user specified string that identifies the command in a command_list. The response for each command will be tagged with corresponding command_id.

Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<response_list status_code="0">
  <get_all_capabilities command_id="get cmd caps id" status_code="0">
    <capability name="CAP_AD_HOC_ITEM_INFO_1"/>
    <capability name="CAP_PLAYER_LOCAL_ITEM_INFO_1"/>
    <capability name="CAP_ADVANCE_TO_1"/>
  </get_all_capabilities>
</response_list>
```

XML Node	Notes
<response_list>	Details ²⁷
status_code	
<get_all_capabilities>	Command for which status is being returned
command_id	is the same as was specified in the request.
status_code	Shared return codes ²⁷
<capability>	Element for holding each capability name.
name	Capability name.

Use case:

This can be used to improve application robustness by ensuring that a specified capability is available before its first use and take appropriate fall back actions if a desired capability is not available.

3.2 CAP_PLAYER_LOCAL_INFO_1

Allows one to set and get player local information. Changes to player local information can immediately affect conditional play items.

3.2.1 Get All Player Local Information

Get all player local information.

Syntax

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<command_list>
  <get_all_player_local_info command_id="all local info"/>
</command_list>
```

XML Node	Notes
<command_list>	Root element for all operations that can be requested
<get_all_player_local_info>	Get Player Local Information command name
command_id	A user specified string that identifies the command in a command_list. The response for each command will be tagged with corresponding command_id.

Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<response_list status_code="0">
  <get_all_player_local_info status_code="0" command_id="all local info">
    <data_player_local_info name="Unit Location">
```

```

        <value>Henley-On-Thames</value>
    <data_player_local_info>
    <data_player_local_info name="Display in">
        <value>Food section</value>
    <data_player_local_info>
</get_all_player_local_info>
</response_list>

```

XML Node	Notes
<response_list>	
status_code	Details ²⁷
<get_all_player_local_info>	Get Player Local Information command name
command_id	Command identifier as specified in the request.
status_code	Shared return codes ²⁷ 100 - Local information file is not valid.
<data_player_local_info>	Container element for player local information item.
name	Player Local Information name.
<value>	Container element for Player Local Information value list. All child text nodes represents a value for the specified name. Multiple values may be associated with a single name.

3.2.2 Get Player Local Information

Get values for a given player local information name.

Syntax

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<command_list>
    <get_player_local_info command_id="cmd_local" name="Unit Location" />
</command_list>

```

XML Node	Notes
<command_list>	Root element for all operations that can be requested
<get_player_local_info>	Get Player Local Information command name
command_id	A user specified string that identifies the command in a command_list. The response for each command will be tagged with corresponding command_id.
name	Player Local Information name.

Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<response_list status_code="0">
  <get_player_local_info status_code="0" command_id="23f">
    <data_player_local_info name="Unit Location">
      <value>Henley-On-Thames</value>
    </data_player_local_info>
  </get_player_local_info>
</response_list>
```

XML Node	Notes
<response_list>	
status_code	Details ^[27]
<get_player_local_info>	Get Player Local Information command name
command_id	Command identifier as specified in the request.
status_code	Shared return codes ^[27] 100 - Local information file is not valid.
<data_player_local_info>	Container element for player local information item.
name	Player Local Information name.
<value>	Container element for Player Local Information value list. All child text nodes represents a value for the specified name. Multiple values may be associated with a single name.

3.2.3 Set Player Local Information

Create or replace player local information.

Syntax

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<command_list>
  <set_player_local_info command_id="cmd_local" name="local info1">
    <data_player_local_info>
      <value>value 1</value>
      <value>value 2</value>
      <value>value 3</value>
    </data_player_local_info>
  </set_player_local_info>
</command_list>
```

XML Node	Notes
<command_list>	Root element for all operations that can be requested
<set_player_local_info>	Set Player Local Information command name
command_id	A user specified string that identifies the command in a command_list. The response for each command will be tagged with corresponding command_id.
name	Player Local Information name.
<data_player_local_info>	Container element for Player Local Information value list
<value>	Container element for each value of Player Local Information.

Response

XML Node	Notes
<response_list>	Details ²⁷
status_code	
<set_player_local_info>	Command for which status is being returned
command_id	is the same as was specified in the request.
status_code	Shared return codes ²⁷ 100 - Local information file is not valid. 101 - Failed to save changes to player local information 102 - Failed to update players about the changes in player local information.

3.3 CAP_ADVANCE_TO_1

Advance all zones or a named zone in a layout to a rendezvous point.

Syntax

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<command_list>
  <advance_to rendezvous_name="rendezvous name" zone_name="zone name" command_id="1"/>
</command_list>
```

XML Node	Notes
<command_list>	Root element for all operations that can be requested
<advance_to>	Advance to command name.
command_id	A user specified string that identifies the command in a command_list. The response for each command will be tagged with corresponding command_id.
rendezvous_name	The name of the rendezvous point to which all zones or a named zone will advance to.
zone_name[Optional]	If specified, only the named zone will advance to the rendezvous point Otherwise all zones in the currently playing layout will receive the command.

Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<response_list status_code="0">
  <advance_to command_id="1" status_code="0"/>
</response_list>
```

XML Node	Notes
<response_list>	Details ²⁷
status_code	
<set_ad_hoc_item_info>	Command for which status is being returned
command_id	is the same as was specified in the request.
status_code	Shared return codes ²⁷

3.4 CAP_AD_HOC_ITEM_1

This capability defines a set of API's to get and set Ad hoc item information for following types of ad hoc items.

- Ad hoc Tables
- Ad hoc Text
- Ad hoc TV
- Ad hoc Radio

get_ad_hoc_item_info

Return code:

100 - Ad hoc item not found.

101 - Ad hoc table data file invalid or corrupt

102 - Command is not supported on named ad hoc item type.

set_ad_hoc_item_info

Return codes:

100 - Ad hoc item not found.

101 - Ad hoc table data file invalid or corrupt

102 - Command is not supported on named ad hoc item type.

103 - Input data for specified ad hoc item is not valid.

104 - Input data type does not match ad hoc item type.

3.4.1 Ad hoc Table

3.4.1.1 Get Ad hoc Table Info

Returns the data for all cells of an Ad hoc table.

Syntax

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<command_list>
  <get_ad_hoc_item_info item_name="adhoc table" command_id="user specified id"/>
</command_list>
```

XML Node	Notes
<command_list>	Root element for all operations that can be requested.
<get_ad_hoc_item_info>	Get ad hoc item command name.
command_id	A user specified string that identifies the command in a command_list. The response for each command will be tagged with corresponding command_id.
item_name	Name of the ad hoc text item of interest.

Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<response_list status_code="0">
  <get_ad_hoc_item_info command_id="user specified id" status_code="0">
    <data_table item_name="adhoc table">
      <row id="1">
        <cell id="A" ad_hoc="yes">Top-Left</cell>
```

```

        <cell id="B" ad_hoc="no">Top-Middle</cell>
        <cell id="C" ad_hoc="no">Top-Right</cell>
    </row>
    <row id="2">
        <cell id="A" ad_hoc="no">Middle-Left</cell>
        <cell id="B" ad_hoc="no">Middle-Middle</cell>
        <cell id="C" ad_hoc="no">Middle-Right</cell>
    </row>
    <row id="3">
        <cell id="A" ad_hoc="no">Bottom-Left</cell>
        <cell id="B" ad_hoc="yes">Bottom-Middle</cell>
        <cell id="C" ad_hoc="yes">Bottom-Right</cell>
    </row>
</data_table>
</get_ad_hoc_item_info>
</response_list>

```

XML Node	Notes
<response_list>	Details ²⁷
status_code	
<get_ad_hoc_item_info>	Command for which the contained response is applicable.
command_id	Command identifier as specified in the request
status_code	Shared return codes ²⁷ 100 Ad hoc item not found.
<data_table>	Container element for ad hoc table data
name	Ad hoc table name for which data was requested.
<row>	Identifies a row in an ad hoc table
id	Unique identifier for an ad hoc table row
<cell>	Identifies a cell in a row. The text nodes represent the content that the table cell will be updated with.
id	Unique identifier for an ad hoc cell in conjunction with row::id
ad_hoc	yes - if the cell content can be updated by an ad hoc user. no - if the cell content cannot be updated by an ad hoc user

3.4.1.2 Set Ad hoc Table Info

Update content of ad hoc cells in an ad hoc table.

Syntax

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<command_list>
  <set_ad_hoc_item_info item_name="adhoc table" command_id="user specified id">
    <data_table>
      <row id="1">
        <cell id="A" >Override</cell>
      </row>
      <row id="3">
        <cell id="B">2:1 Unicode - €ßß«</cell>
        <cell id="C">2:2 Bottom-Right Red</cell>
      </row>
    </data_table>
  </set_ad_hoc_item_info>
</command_list>
```

XML Node	Notes
<command_list>	Root element for all operations that can be requested
<set_ad_hoc_item_info>	Set ad hoc item command name
command_id	A user specified string that identifies the command in a command_list. The response for each command will be tagged with corresponding command_id.
item_name	Name of the ad hoc table that needs to be updated
<data_table>	Container element for ad hoc table data
<row>	Uniquely identifies a row in an ad hoc table
id	Identifier of an ad hoc table row
<cell>	Uniquely identifies a cell in a row. The text nodes represent the new data for this cell.
id	Cell Identifier

Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<response_list status_code="0">
  <set_ad_hoc_item_info command_id="user specified id" status_code="0"/>
</response_list>
```

XML Node	Notes
<response_list>	Details ²⁷
status_code	
<set_ad_hoc_item_info>	Command for which status is being returned
command_id	is the same as was specified in the request.
status_code	Numeric value to indicate success or failure of requested operation.

3.4.2 Ad hoc Text

3.4.2.1 Get Ad hoc Text Info

Get contents of an ad hoc text item.

Syntax

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<command_list>
  <get_ad_hoc_item_info command_id="user specified id" item_name="adhoc text"/>
</command_list>
```

XML Node	Notes
<command_list>	Root element for all operations that can be requested
<get_ad_hoc_item_info>	Get ad hoc item command name
command_id	A user specified string that identifies the command in a command_list. The response for each command will be tagged with corresponding command_id.
item_name	Name of the ad hoc text item of interest

Response

Plain text item

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<response_list status_code="0">
  <get_ad_hoc_item_info command_id="user specified id" status_code="0">
    <data_text_plain item_name="adhoc text">Ad Hoc control test data</data_text_plain>
  </get_ad_hoc_item_info>
</response_list>
```

Rich text item

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<response_list status_code="0">
  <get_ad_hoc_item_info command_id="user specified id" status_code="0">
    <data_text_rich item_name="adhoc text">Ad Hoc control test data</data_text_rich>
  </get_ad_hoc_item_info>
</response_list>
```

XML Node	Notes
<response_list>	
status_code	Details ²⁷
<get_ad_hoc_item_info>	Command for which the contained response is applicable.
command_id	Command identifier as specified in the request
status_code	
<data_text_plain>	Container element for ad hoc text item if the text type is plain
item_name	Name of the ad hoc text item of interest
<data_text_rich>	Container element for ad hoc text item if the text type is rich
item_name	Name of the ad hoc text item of interest

3.4.2.2 Set Ad hoc Text Info

Update the contents of an ad hoc text item.

Syntax**Rich Ad hoc Text Item**

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<command_list>
  <set_ad_hoc_item_info command_id="user specified id" item_name="ad hoc Control">
    <data_text_rich>Ad Hoc control test data: modified</data_text_rich>
  </set_ad_hoc_item_info>
</command_list>
```

Plain Ad hoc Text Item

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<command_list>
  <set_ad_hoc_item_info command_id="user specified id" item_name="ad hoc Control">
    <data_text_plain>Ad Hoc control test data: modified</data_text_plain>
  </set_ad_hoc_item_info>
```

```
</command_list>
```

XML Node	Comment
<command_list>	Root element for all operations that can be requested
<set_ad_hoc_item_info>	Set ad hoc item command name
command_id	A user specified string that identifies the command in a command_list. The response for each command will be tagged with corresponding command_id.
item_name	Name of the ad hoc text item of interest
<data_text_rich>	Container element for ad hoc text item if the text type is rich. The text nodes represent the content that ad hoc item will be updated with.
<data_text_plain>	Container element for ad hoc text item if the text type is plain. The text nodes represent the content that ad hoc item will be updated with.

Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<response_list status_code="0">
  <set_ad_hoc_item_info command_id="user specified id" status_code="0"/>
</response_list>
```

Attribute	Notes
<response_list>	Details ²⁷
status_code	
<set_ad_hoc_item_info>	Command for which status is being returned
command_id	is the same as was specified in the request.
status_code	Numeric value to indicate success or failure of requested operation.

3.4.3 Ad hoc TV

3.4.3.1 Get Ad hoc TV Info

Get a list of channels available for an ad hoc tv item along with the currently selected channel.

Syntax


```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<command_list>
  <get_ad_hoc_item_info command_id="user specified id" item_name="adhoc tv item"/>
</command_list>
```

```
</command_list>
```

XML Node	Notes
<command_list>	Root element for all operations that can be requested
<get_ad_hoc_item_info>	Get ad hoc item command name
command_id	A user specified string that identifies the command in a command_list. The response for each command will be tagged with corresponding command_id.
item_name	Name of the ad hoc text item of interest

Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<response_list status_code="0">
  <get_ad_hoc_item_info status_code="0" command_id="1">
    <data_tv item_name="ad hoc tv">
      <choice selected="no">BBC NEWS 24 (S) Stream</choice>
      <choice selected="yes">BBC ONE (S) Stream</choice>
      <choice selected="no">AUX 1 Stream</choice>
      <choice selected="no">CBBC Channel (S) Stream</choice>
      <choice selected="no">BBC TWO (S) Stream</choice>
    </data_tv>
  </get_ad_hoc_item_info>
</response_list>
```

XML Node	Notes
<response_list>	Details 
status_code	
<get_ad_hoc_item_info>	Command for which the contained response is applicable.
command_id	Command identifier as specified in the request.
status_code	
<data_tv>	Container element for ad hoc tv item.
item_name	Name of ad hoc item of interest.
<choice>	Container element for TV choice. The text node represents one channel
selected	yes - if the channel is currently selected and available. no - otherwise.

3.4.3.2 Set Ad hoc TV Info

Change the selection in Ad hoc TV item.

Syntax

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<command_list>
  <set_ad_hoc_item_info command_id="user specified id" item_name="ad hoc tv Control">
    <data_tv>Channel 1</data_tv>
  </set_ad_hoc_item_info>
</command_list>
```

XML Node	Notes
<command_list>	Root element for all operations that can be requested
<set_ad_hoc_item_info>	Get ad hoc item command name
command_id	A user specified string that identifies the command in a command_list. The response for each command will be tagged with corresponding command_id.
item_name	Name of the ad hoc text item of interest
<data_tv>	Container element for name of the new TV channel

Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<response_list status_code="0">
  <set_ad_hoc_item_info command_id="user specified id" status_code="0"/>
</response_list>
```

XML Node	Notes
<response_list>	Details ²⁷
status_code	
<set_ad_hoc_item_info>	Command for which status is being returned
command_id	is the same as was specified in the request.
status_code	Numeric value to indicate success or failure of requested operation.

3.4.4 Ad hoc Radio

3.4.4.1 Get Ad hoc Radio

Get a list of channels available for an ad hoc tv item along with the currently selected channel.

Syntax

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<command_list>
  <get_ad_hoc_item_info command_id="user specified id" item_name="ad hoc radio example"/>
</command_list>
```

XML Node	Comment
<command_list>	Root element for all operations that can be requested
<get_ad_hoc_item_info>	Get ad hoc item command name
command_id	A user specified string that identifies the command in a command_list. The response for each command will be tagged with corresponding command_id.
item_name	Name of the ad hoc text item of interest

Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<response_list status_code="0">
  <get_ad_hoc_item_info status_code="0" command_id="1">
    <data_radio item_name="ad hoc radio example">
      <choice selected="no">BBC Radio 1 (S) (Stream)</choice>
      <choice selected="yes">BBC 6 Music (S) (Stream)</choice>
      <choice selected="no">TECH TVG Absolute Radio (Stream)</choice>
    </data_radio>
  </get_ad_hoc_item_info>
</response_list>
```

XML Node	Notes
<response_list>	
status_code	Details ²⁷
<get_ad_hoc_item_info>	Command for which the contained response is applicable.
command_id	Command identifier as specified in the request.
status_code	
<data_radio>	Container element for ad hoc tv item.
item_name	Name of ad hoc item of interest.
<choice>	Container element for TV choice. The text node represents one channel
selected	yes - if the channel is currently selected and available. no - otherwise.

3.4.4.2 Set Ad hoc Radio

Change the selection in Ad hoc Radio item.

Syntax

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<command_list>
  <set_ad_hoc_item_info command_id="user specified id" item_name="ad hoc radio Control">
    <data_radio>Channel 1</data_radio>
  </set_ad_hoc_item_info>
</command_list>
```

XML Node	Comment
<command_list>	Root element for all operations that can be requested
<set_ad_hoc_item_info>	Get ad hoc item command name
command_id	A user specified string that identifies the command in a command_list. The response for each command will be tagged with corresponding command_id.
item_name	Name of the ad hoc text item of interest
<data_radio>	Container element for name of the new TV channel

Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<response_list status_code="0">
  <set_ad_hoc_item_info command_id="user specified id" status_code="0"/>
</response_list>
```

Attribute	Notes
<response_list>	Details ²⁷
status_code	
<set_ad_hoc_item_info>	Command for which status is being returned
command_id	is the same as was specified in the request.
status_code	Numeric value to indicate success or failure of requested operation.

3.5 Shared Return Codes

These return codes apply to all commands in a command list. Individual commands may have their own errors that start from 100.

- 0 - The command executed successfully.
- 1 - There was an error while executing the command.
- 2 - The command was not executed due to error(s) in a previous command.

3.6 Response List

XML Node	Notes
<response_list>	Root element for all responses
status_code	<p>Indicates an overall status of all commands in the command list.</p> <ul style="list-style-type: none"> 0 - All commands succeeded 1 - Invalid XML 2 - Partial Failure- At least one command failed. 3 - command_list contained a mix of state changing and non-state changing commands. 4 - The request was not sent with correct HTTP headers. 99 - For all other errors.

Error format

```
<response_list status_code="1">
  <CMD_NAME_PLACEHOLDER command_id="user specified id" status_code="100">
    <status>ERROR_MESSAGE_PLACEHOLDER</status>
  </CMD_NAME_PLACEHOLDER>
```

```
</response_list>
```

CMD_NAME_PLACEHOLDER - Represents the name of the command that failed.

ERROR_MESSAGE_PLACEHOLDER - A detailed message about the error.

Part



IV

XML API version 1

4. XML API version 1

Enter topic text here.

4.1 External XML API

The calling application must handle HTTP 401 status code and authenticate with the request URI.
All external XML API can also be used from media running within the iCOMPEL HTML Player.

Following API are available.

- [Activate Schedule](#)^[34]
- [Channel Activate](#)^[33]
- [Channel Publish](#)^[33]
- [Get Channel Manager Report](#)^[31]

4.1.1 Channel Manager

This section contains information that enables building very flexible content publishing and subscription network using Black Box digital signage.

4.1.1.1 Channel Manager Status

Function

A detailed report of channel manager state for the currently active channel on a publisher and subscribed channel for a subscriber.

In conjunction with other Channel Manager XML API, these can help build very flexible channel publishing work flows.

Requires

An iCOMPEL user with '**Monitor**' access.

An application that can process HTTP GET requests with digest or basic authentication. Versions prior to 8.0 support basic authentication only.

Syntax

HTTP Get the URI `/status/channel_manager.xml`

Publisher

The following is returned:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<channel_manager_status role="publisher">
<!-- Publisher -->
<channel_control_file channel_name="Channel 2" channel_unique_id="2010-10-04 15:47:18" channel_href="file:
<channel_control_publish state="Publishing - Copying channel files" state_code="publishing"/>
  <channel_file_list>
    <channel_file href="file:control/channel_control.xml" channel_url="file:channel/2/control/channel_c
    <channel_file href="file:control/colours.xml" channel_url="file:channel/2/control/colours.xml.b1c28
```

```

        <channel_file href="file:media/show/office/back/4x3/Red.png" size="402437" last_error="pending"/>
        <channel_file href="file:media/show/soundtrack/the_power_behind_full.mp3" size="4215118" last_error="pending"/>
    </channel_file_list>
</channel_manager_status>

```

The 'state_code' attribute indicates the current status of channel publish and can take any of 'not_published', 'publishing' or 'published' values.

Subscriber

The following is returned:

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<channel_manager_status role="subscriber">
    <channel_state_subscribe>
        <active_channel state="activated" channel_name="channel 1" channel_id="2010-09-27 11:32:21" act
        <pending_channel state="unknown" transfer_start_time="2010-10-04 16:15:29" transfer_end_time="2010-10-04 16:15:29"/>
    </channel_state_subscribe>
    <channel_control_subscribe state="Idle" channel_control_url="ftp://ftpserver/channel/channel_id/control/channel_control.xml" size="9984"/>
    <channel_file_list>
        <channel_file href="ftp://testproxy/channel/14533/control/channel_control.xml" size="9984"/>
        <channel_file href="file:control/ad_hoc_control_settings.xml" size="278"/>
        <channel_file href="file:control/ad_hoc_form_settings.xml" size="1066"/>
        <channel_file href="file:control/colours.xml" size="914"/>

        <channel_file href="file:control/rs232_settings_default.xml" size="1066"/>
    </channel_file_list>

</channel_manager_status>

```

The 'active_channel' attribute can be either 'activated' or 'never_activated'.

Compatibility

Introduced in 7.4.

4.1.1.2 Get Channel Manager Report

Function

Provide information about iCOMPEL channel manager to determine mode and state of all channels.

On a publisher, verify the status of channel publish completion by comparing the `total_size` with `total_transferred_size`.

On a subscriber, verify the status of channel subscription completion by comparing the `total_size` with `total_transferred_size`.

In conjunction with [Channel Activate](#)^[33] and [Channel Publish](#)^[33], provides a mechanism to build flexible channel publishing work flows.

Syntax

Example:

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

```

```
<get_channel_manager_report/>
```

Examples

Standalone mode

Example Return XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<channel_manager_report>
  <channel_manager_standalone/>
</channel_manager_report>
```

Publisher Mode

Example Return XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<channel_manager_report>
  <channel_manager_publisher
    multichannel_id="1" multichannel_active="yes"
    channel_name="Channel 1" channel_id="2010-09-27 11:32:21"
    channel_type="ftp" channel_href="ftp://ftpserver/channel/channel_folder"
    transfer_start_time="2010-09-27 11:32:26"
    last_connect_time="2010-09-27T11:32:26+0100"
    total_files="72" total_transferred_files="63"
    total_size="225393921" total_transferred_size="121803417"
  />
  <channel_manager_publisher
    multichannel_id="2" multichannel_active="no"
    channel_name="Channel 2"
    channel_type="local_disk" channel_href="file:channel/2"
  />
</channel_manager_report>
```

Subscriber Mode

Example Return XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<channel_manager_report>
  <channel_manager_subscriber
    multichannel_id="1" multichannel_active="yes" last_poll_time="2010-09-30T15:15:35+0100"
    channel_type="ftp" channel_href="ftp://ftpserver/channel/channel_folder/"
    pending_channel_name="channel1"
    pending_channel_id="2010-09-27 11:32:21"
    transfer_start_time="2010-09-30 15:15:36"
    next_connect_time="2010-09-30T16:15:36+0100"
    last_connect_time="2010-09-30T15:15:35+0100"
    total_files="72" total_transferred_files="59"
    total_size="225393921" total_transferred_size="38125373"/>
  </channel_manager_report>
```


Compatibility

Introduced in 7.4.

4.1.1.3 Channel Publish

Function

A mechanism to publish the currently active channel.

Allow external applications to control when channel gets published and when subscribers can activate the new channel.

The currently active channel is published. A channel can be activated by using [Channel Activate](#)^[33] API.

Requires

An application that can send HTTP POST requests with digest and basic authentication.

The user must also have '**XML Control**' permission.

Syntax

Publish a channel and set to be activated immediately:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<channel_publish/>
```

Publish the currently active channel with user specified 'activate_after'.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<channel_publish activate_after="2010-02-13 01:20:00"/>
```

Return Status

HTTP Status code 200 on success, 500 otherwise.

Compatibility

Introduced in 7.4.

4.1.1.4 Channel Activate

Function

A mechanism to activate a channel on a iCOMPEL Publisher.

Allow external applications to control channel activation. In conjunction with [Channel Publish](#)^[33] and [Get Channel Manager Report](#)^[31], allows a single publisher to publish to multiple channels with verifiable publish completion.

Syntax

Activate a channel with a specified Channel id:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<channel_activate multichannel_id="2">
```

The 'multichannel_id' attribute can be extracted from [Get Channel Manager Report](#)^[31]

Return Status

HTTP Status code 200 on success, 500 otherwise.

Compatibility

Introduced in 7.4.

4.1.2 Schedule

A calendar of what layouts can play during a given time period.

4.1.2.1 Schedule Status

Function

A mechanism to get important information about schedule status.

This information can be used to determine schedules that can be used with [Activate Schedule](#)^[34] command.

Requires

An iCOMPEL user with **Monitor** access permission.

An application that can process HTTP GET requests with digest or basic authentication. Versions prior to 8.0 support basic authentication only.

Syntax

HTTP Get the URI `/status/schedule.xml`

The returned XML has an element named `'schedule_config'` that contains the details of currently available schedules.

Example Return XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<status>
  <schedule_status>
    ~snip~
  </schedule_status>
  <schedule_config href="file:control/channel/active/schedule.xml">
    <schedule_override name="special" href="file:control/channel/active/schedule/override/overr
  </schedule_config>
</status>
```

The `'schedule_override'` attribute can be used with [Activate Schedule](#)^[34]

Compatibility

Introduced in 7.4.

4.1.2.2 Activate Schedule

Function

A mechanism to activate a specific schedule from available schedules.

Allow external applications to control schedule overrides.

Syntax

Override with a special schedule:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<activate_schedule href="file:control/channel/active/schedule/override/override_evening.xml" />
```

Revert back to normal schedule:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<activate_schedule href="file:control/channel/active/schedule.xml" />
```

For schedules that can be used with 'activate_schedule', refer [Schedule Status](#)³⁴

Return Status

HTTP Status code 200 on success, 500 otherwise.

Compatibility

Introduced in 7.4.

4.2 HTML Player Services

Overview

The iCOMPEL HTML player displays HTML and Adobe Flash based play list items. These items can execute scripts which interact with iCOMPEL features.

In order to interact with the iCOMPEL, XML Web Services must be enabled for the HTML player and for each HTML play list item individually. The XML Web services are disabled by default.

Enable XML Web services for HTML Player

In iCOMPEL Web UI:

1. Visit **Setup > Player Setup > Players** page
2. Choose **'Allow XML web services to access external sites'** to allow the HTML Player to access external sites.
Choose **'Do not allow XML web services to access external sites'** otherwise.

Enable XML Web services for HTML Play list item

Each individual HTML play list item must also have 'Page Services' option enabled.

To enable Page Services:

1. Click 'Edit' on an HTML play list item
2. Select 'Style Tab'.
3. Ensure 'Page Services' is checked and 'Save Changes'.

4.2.1 Get RS-232 Buffer

Function

A mechanism to read and optionally clear RS-232 buffered input.

Allow external applications to query and optionally clear the RS-232 buffer. This allows devices with RS-232 communication interface to dynamically interact with iCOMPEL Digital Signage platform

Syntax

Get RS-232 Buffered Data without clearing the buffer:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<get_rs232_buffered_data clear_buffer="no"/>
```

Get RS-232 Buffered Data clearing the buffer.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<get_rs232_buffered_data clear_buffer="yes"/>
```

Return Status

HTTP Status code 200 on success, 500 otherwise.

Example

If the buffer contained **'Welcome to Digital Signage'**, the output will be

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<rs232_buffered_data>57656c63666d65207465204469676974616c205369676e616765</rs232_buffered_data>
```

The data element is hex encoding of **'Welcome to Digital Signage'**.

If the buffer is empty:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<rs232_buffered_data></rs232_buffered_data>
```

Compatibility

Introduced in 8.0.

4.2.2 Transmit RS-232 Data

Function

A mechanism to send RS-232 output from a iCOMPEL.

Enables one to write dynamic web applications that can drive external RS-232 devices.

Syntax

The data must be encoded in hex format for transmission.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<transmit_rs232_data>61636b206669727374206d7367</transmit_rs232_data>
```

Return Status

HTTP Status code 200 on success, 500 otherwise.

Compatibility

Introduced in 8.0.

4.2.3 HTTP GET Request

Function

Allow HTML pages being played in iCOMPEL to access resources that might otherwise be unavailable due to cross-domain security policies in HTML Player.

Enables building complex mash ups for digital signage where content may need to be fetched from different sources while ensuring the security of the signage system.

This API uses XML Services proxy, if configured.

Syntax

Example XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<http_request href="http://www.onelan.co.uk" method="get"
              username="fakeuser"
              password="fakepassword">
  <header name="accept-language" value="en-GB" />
</http_request>
```

The header names are case-insensitive.

You may set the following HTTP headers to be forwarded to the remote server

Header Name	Description	Default
Accept	acceptable content type	text/*
Accept-language	acceptable language for response	en-GB
Cookie	an HTTP cookie sent by server in a previous call	
User-agent	application type requesting the URI	Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.8) Gecko/20051111 Firefox/1.5

Return Status

The Return Status will be the same as if the user had accessed the resource through a browser.

Applications may check for 'Content-type' to determine the returned resource.

Compatibility

Introduced in 8.0.

4.2.4 HTTP POST Request

Function

Allow HTML pages played in iCOMPEL to submit web forms and post data to URI using HTTP POST method.

This will enable the HTML player to play web pages that require HTML form based login. A custom HTML page can be built to HTTP Get the new page, submit a login form, setup a session and display content that otherwise would not have been possible.

This API uses XML Services proxy, if configured.

Syntax

The following XML shall be posted to '/XML' with 'Content-type' set to 'text/xml':

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<http_request href="http://www.onelan.co.uk"
    username="fakeuser"
    password="fakepassword"
    method="post">
    <header name='language' value='en-GB' />
    <content content_type="x-www-form-urlencoded">&amp;query_id=2</content>
</http_request>
```

Return Status

HTTP Status code and content as returned by remote server on success, 500 otherwise.

Compatibility

Introduced in 8.0.

4.3 Sample Code

Below is sample C# code which will Post an Advance To message.

```
using System;
using System.Text;
using System.Net;
using System.IO;
using System.Xml;
using System.Security;

namespace TestNTBPost
{
    class NTBPost
    {
        static public void postAdvanceTo(string rendezvous_name, string hostname, string username, string password)
        {
            try
            {
                Console.WriteLine( string.Format(
                    "Info: Post advance to: {0} to NTB: {1}",
                    rendezvous_name, hostname));

                string advance_to_message = string.Format(
```

```

        "<?xml version=\"1.0\" encoding=\"UTF-8\" standalone=\"yes\"?>\n
          <advance_to rendezvous_name=\"{0}\"/>",
        SecurityElement.Escape(rendezvous_name)
    );

    string url = string.Format("http://{0}/XML/", hostname);
    Uri uri = new Uri(url);

    HttpWebRequest request = WebRequest.Create(uri) as HttpWebRequest;
    request.Proxy = null;
    request.KeepAlive = true;
    request.PreAuthenticate = true;
    request.ServicePoint.Expect100Continue = false;

    CredentialCache cc = new CredentialCache();
    cc.Add(uri, "Basic", new NetworkCredential(username, password));
    cc.Add(uri, "Digest", new NetworkCredential(username, password));
    request.Credentials = cc;

    request.Method = "POST";

    // Create POST data and convert it to a byte array.
    byte[] byteArray = Encoding.UTF8.GetBytes(advance_to_message);
    request.ContentType = "text/xml; charset=utf-8";
    request.ContentLength = byteArray.Length;
    using (Stream dataStream = request.GetRequestStream())
    {
        dataStream.Write(byteArray, 0, byteArray.Length);
    }

    // Get the response.
    using (HttpWebResponse response = request.GetResponse() as HttpWebResponse)
    {
        if (response.StatusCode == HttpStatusCode.OK)
        {
            Console.WriteLine("Info: Successfully posted AdvanceTo message");
        }
        else
        {
            Console.WriteLine(string.Format(
                "Error: Failed to post message. HTTP response - {0}",
                response.StatusDescription));

            using (Stream dataStream = response.GetResponseStream())
            {
                using (StreamReader reader = new StreamReader(dataStream))
                {
                    string responseFromServer = reader.ReadToEnd();
                    Console.WriteLine(string.Format(
                        "Error: Server responded: {0}",
                        responseFromServer));
                }
            }
        }
    }
}

```

```
        }  
    }  
    catch (WebException ex)  
    {  
        Console.WriteLine(string.Format(  
            "Error: WebException when posting message: {0}",  
            ex.Message));  
    }  
}  
}  
}
```


Index

- A -

Activate Schedule 34

- C -

Channel Activate 33

Channel Manager 30

Channel Manager Status 30

Channel Publish 33

- G -

Get Channel Manager Report 31

Get RS-232 Buffer 35

- H -

HTML Player Services 35

HTTP GET Request 37

HTTP POST Request 38

- P -

Prerequisites 9

- S -

Schedule 34

Schedule Status 34

- T -

Transmit RS-232 Data 36

Black Box Tech Support: FREE! Live. 24/7.

Tech support the
way it should be.



Great tech support is just 30 seconds away at 724-746-5500 or blackbox.com.



About BlackBox

Black Box Network Services is your source for more than 118,000 networking and infrastructure products. You'll find everything from cabinets and racks and power and surge protection products to media converters and Ethernet switches all supported by free, live 24/7 Tech support available in 30 seconds or less.

© Copyright 2010. All rights reserved. Black Box® and the Double Diamond logo are registered trademarks of BB Technologies, Inc. Any third-party trademarks appearing in this manual are acknowledged to be the property of their respective owners.

ICOMP, rev. 1 (build: V8.1.0 (23484))